UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/713,656	11/14/2003	John E. Howe	1138-003	9852

34060 7590 03/15/2010

MICHAEL N. HAYNES
1341 HUNTERSFIELD CLOSE
KESWICK, VA 22947

EXAMINER	
BATURAY, ALICIA	
ART UNIT	PAPER NUMBER
2446	
MAIL DATE	DELIVERY MODE
03/15/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/713,656	**Applicant(s)** HOWE ET AL.	
	Examiner Alicia Baturay	**Art Unit** 2446	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 January 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-9,11-14,17-26,28-31,34-38 and 40-48* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-9, 11-14, 17-26, 28-31, 34-38 and 40-48* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *05 January 2010* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is in response to the amendment filed 05 January 2010.

2. Claims 1, 5, 6, 17-19, 23, 24, 34, 35, 41, 43 and 44 were amended.

3. Claims 10, 15, 16, 27, 32, 33 and 39 were cancelled.

4. Claims 1-9, 11-14, 17-26, 28-31, 34-38 and 40-48 are pending in this Office Action.

Response to Amendment

5. The objections to the informal drawings were addressed and are withdrawn.

6. The objection to Fig. 3 was addressed and is withdrawn.

7. The objection to the specification regarding trademark usage was addressed and is withdrawn.

8. Applicant's amendments and arguments with respect to claims 1-9, 11-14, 17-26, 28-31, 34-38 and 40-48 filed on 05 January 2010 have been fully considered but they are deemed to be moot in view of the new grounds of rejection.

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

10. Claims 1-9, 17-26, 34 and 44-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ignatius et al. (U.S. 7,209,972) and further in view of Ganger et al. ("Fast and Flexible Application-Level Networking on Exokernel Systems").

Ignatius teaches the invention substantially as claimed including a storage and data management system establishes a data transfer pipeline between an application and a storage media using a source data mover and a destination data mover. The data movers are modular software entities which compartmentalize the differences between operating systems and media types. In addition, they independently interact to perform encryption, compression, etc., based on the content of a file as it is being communicated through the pipeline. Headers and chunking of data occurs when beneficial without the application ever having to be aware. Faster access times and storage mapping offer enhanced user interaction (see Abstract).

11. With respect to claim 1, Ignatius teaches a method for reducing processor cycles required to send data over a communication link in packets having a packet size, the method comprising: sending a write call, from an application running in a first context, comprising a first destination and pointing to a first quantity of data stored in virtual memory destined for the first destination (Ignatius, col. 15, line 55 – col. 16, line 15) and a second destination and pointing to a second quantity of data stored in virtual memory destined for the second destination (Ignatius, col. 16, lines 16-27), both the first quantity of data and the second quantity of data greater than said packet size to a driver (Ignatius, col. 16, lines 46-60), a network interface controller (Ignatius, Fig. 1, elements 76 and 78; col. 4, line 62 – col. 5, line

13), running in a second context through a socket (Ignatius, col. 16, lines 46-60); and generating, by the driver, a plurality of first packets less than or equal to the packet size destined for the first destination from the first quantity of data and the plurality of second packets less than or equal to the packet size destined for the second destination from the second quantity of data (Ignatius, col. 16, lines 46-60), the first and second quantities of data located at the translated physical memory locations (Ignatius, col. 3, lines 57-61).

Ignatius does not explicitly teach performing a zero-copy write.

However, Ganger teaches performing a zero-copy write translating the virtual memory locations for the first and second quantities of data to physical memory locations (Ganger, page 71, 2^{nd} full paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ignatius in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

12. With respect to claims 2 and 20, Ignatius teaches the invention described in claim 1, including a method further comprising grouping data from a plurality of streams into said write call (Ignatius, col. 15, lines 11-13).

13. With respect to claims 3 and 21, Ignatius teaches the invention described in claim 2, including a method wherein said grouped data comprises all data from said plurality of streams to be sent in a time interval (Ignatius, col. 15, lines 11-13).

14. With respect to claims 4 and 22, Ignatius teaches the invention described in claim 4, including a method wherein the time interval is selected based on a bit rate of at least one stream (Ignatius, col. 10, lines 54-63).

15. With respect to claims 5 and 23, Ignatius teaches the invention described in claim 1, including a method wherein the virtual memory comprises a translation buffer, the driver having a translation mapping from the virtual memory of the translation buffer to physical memory (Ignatius, col. 3, lines 57-61).

16. With respect to claims 6 and 24, Ignatius teaches the invention described in claim 5, including a method for reducing processor cycles required to send data over a communication link in packets having a packet size, the method comprising: sending a write call, from an application running in a first context, comprising a first destination and pointing to a first quantity of data stored in virtual memory destined for the first destination (Ignatius, col. 15, line 55 – col. 16, line 15) and a second destination and pointing to a second quantity of data stored in virtual memory destined for the second destination (Ignatius, col. 16, lines 16-27), both the first quantity of data and the second quantity of data greater than said packet size to a driver (Ignatius, col. 16, lines 46-60), a network interface controller (Ignatius, Fig. 1, elements 76 and 78; col. 4, line 62 – col. 5, line 13), running in a second context through a socket (Ignatius, col. 16, lines 46-60); and generating, by the driver, a plurality of first packets less than or equal to the packet size destined for the first destination from the first quantity of data and the plurality of second packets less than or equal to the packet size

destined for the second destination from the second quantity of data (Ignatius, col. 16, lines 46-60), the first and second quantities of data located at the translated physical memory locations (Ignatius, col. 3, lines 57-61) and the method wherein the write comprises translating the virtual memory location of the translation buffer to the physical memory location using the translation mapping of the driver (Ignatius, col. 4, line 46 – col. 5, line 13).

Ignatius does not explicitly teach performing a zero-copy write.

However, Ganger teaches performing a zero-copy write translating the virtual memory locations for the first and second quantities of data to physical memory locations (Ganger, page 71, 2nd full paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ignatius in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

17. With respect to claims 7 and 25, Ignatius teaches the invention described in claim 1, including a method further comprising generating an interrupt only after a last packet of said plurality of packets is transmitted to said communication link (Ignatius, col. 13, lines 23-25).

18. With respect to claims 8 and 26, Ignatius teaches the invention described in claim 1, including a method wherein said packet size is a maximum packet size allowable by the communication link (Ignatius, col. 4, line 46 – col. 5, line 13).

19. With respect to claim 9, Ignatius teaches the invention described in claim 1, including a method wherein the communication link comprises a network (Ignatius, col. 4, line 62 – col. 5, line 13).

20. With respect to claims 17 and 34, Ignatius teaches the invention described in claim 1, including a method wherein said write call comprises a write vector (Ignatius, col. 15, lines 11-13) with entries for the first destination, the first data destined for the first destination (Ignatius, col. 15, line 65 - col. 16, line 15), the second destination and the second data destined for the second destination (Ignatius, col. 16, lines 16-27).

21. With respect to claim 18, Ignatius teaches the invention described in claim 1, including a method further comprising generating a single header comprising header information for a plurality of protocol layers and sending the single header to a queue for the NIC (Ignatius, col. 17, lines 12-29).

22. With respect to claim 19, Ignatius teaches a computer program product for sending data over a communications link in packets having a packet size, the computer program product comprising: a computer-readable medium comprising a program module, the program module including instructions for: receiving a write call from an application running in a first context (Ignatius, col. 6, line 48 – col. 7, line 9), comprising a first destination and pointing to a first quantity of data stored in virtual memory destined for the first destination (Ignatius, col. 15, line 65 - col. 16, line 15) and a second destination and pointing to a second quantity

of data stored in virtual memory destined for the second destination (Ignatius, col. 16, lines 16-27), both the first quantity of data and the second quantity of data greater than said packet size through a socket (Ignatius, col. 16, lines 46-60); and generating, by the driver, a plurality of first packets less than or equal to the packet size destined for the first destination from the first quantity of data (Ignatius, col. 16, lines 46-60) and a plurality of second packets less than or equal to the packet size destined for the second destination from the second quantity of data (Ignatius, col. 17, lines 12-29), the first and second quantities of data located at the translated physical memory locations (Ignatius, col. 3, lines 57-61).

Ignatius does not explicitly teach performing a zero-copy write.

However, Ganger teaches performing a zero-copy write translating the virtual memory locations for the first and second quantities of data to physical memory locations (Ganger, page 71, 2nd full paragraph) and the write call received at a driver running a second context (Ganger, page 54, 2.2 Application-Level Networking, 1st paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ignatius in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

23. With respect to claim 44, Ignatius teaches a system for sending data across a network, the system comprising: a computer running an application in a first context configured to send through a socket (Ignatius, col. 6, line 48 – col. 7, line 9), the write call comprising a first destination and pointing to a first quantity of data stored in virtual memory destined for the

first destination (Ignatius, col. 15, line 55 – col. 16, line 15) and a second destination and pointing to a second quantity of data stored in virtual memory destined for the second destination (Ignatius, col. 16, lines 16-27), both the first quantity of data and the second quantity of data greater than a packet size (Ignatius, col. 16, lines 46-60), a network interface controller (Ignatius, Fig. 1, elements 76 and 78; col. 4, line 62 – col. 5, line 13); the driver configured to generate a plurality of first packets less than or equal to the packet size destined for the first destination from the first quantity of data and a plurality of second packets less than or equal to the packet size destined for the second destination from the second quantity of data (Ignatius, col. 16, lines 46-60), the first and second quantities of data located at the translated physical memory locations (Ignatius, col. 16, lines 46-60); and a downstream device adapted to receive at least one of said packets (Ignatius, col. 12, lines 45-52).

Ignatius does not explicitly teach performing a zero-copy write.

However, Ganger teaches the application further configured to perform a zero-copy write to translate the virtual memory locations for the first and second quantities of data to physical memory locations (Ganger, page 71, 2nd full paragraph) and to a driver running in a second context (Ganger, page 54, 2.2 Application-Level Networking, 1st paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ignatius in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

24. With respect to claim 45, Ignatius teaches the invention described in claim 44, including a system further comprising a network switch coupled to the computer and the downstream device, adapted to receive at least one of said packets (Ignatius, col. 6, line 42-47) and route the received packet to the downstream device (Ignatius, col. 15, line 65 – col. 16, line 15).

25. With respect to claim 46, Ignatius teaches the invention described in claim 44, including a system wherein the downstream device comprises a down stream device has a timing requirement for receipt of data (Ignatius, col. 16, lines 46-60).

26. With respect to claim 47, Ignatius teaches the invention described in claim 45, including a system further comprising a plurality of said computers in communication with the network switch (Ignatius, col. 15, line 65 – col. 16, line 15).

27. With respect to claim 48, Ignatius teaches the invention described in claim 44, including a system further comprising a plurality of downstream devices in communication with the network switch (Ignatius, col. 15, line 65 – col. 16, line 15).

28. Claims 11-14, 28-31, 35-38 and 40-43 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ignatius in view of Ganger and further in view of Haddock et al. (U.S. 6,104,700).

29. With respect to claims 11 and 28, Ignatius teaches the invention described in claim 1, including a method for reducing processor cycles required to send data over a communication link in packets having a packet size, the method comprising: sending a write call, from an application running in a first context, comprising a first destination and pointing to a first quantity of data stored in virtual memory destined for the first destination (Ignatius, col. 15, line 55 – col. 16, line 15) and a second destination and pointing to a second quantity of data stored in virtual memory destined for the second destination (Ignatius, col. 16, lines 16-27), both the first quantity of data and the second quantity of data greater than said packet size to a driver (Ignatius, col. 16, lines 46-60), a network interface controller (Ignatius, Fig. 1, elements 76 and 78; col. 4, line 62 – col. 5, line 13), running in a second context through a socket (Ignatius, col. 16, lines 46-60); and generating, by the driver, a plurality of first packets less than or equal to the packet size destined for the first destination from the first quantity of data and the plurality of second packets less than or equal to the packet size destined for the second destination from the second quantity of data (Ignatius, col. 16, lines 46-60), the first and second quantities of data located at the translated physical memory locations (Ignatius, col. 3, lines 57-61).

Ignatius does not explicitly teach performing a zero-copy write.

However, Ganger teaches performing a zero-copy write translating the virtual memory locations for the first and second quantities of data to physical memory locations (Ganger, page 71, 2nd full paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ignatius in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

The combination of Ignatius and Ganger does not explicitly teach use of a multimedia data file.

However, Haddock teaches a method wherein said quantity of data comprises at least a portion of a multimedia data file (Haddock, col. 8, lines 45-47).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the combination of Ignatius and Ganger in view of Haddock in order to enable transmitting at least one packet to the second destination before transmitting a second packet for the first destination. One would be motivated to do so in order to employ a weighted fair queuing delivery schedule which shares available bandwidth so that high priority traffic is usually sent first, but low priority traffic is still guaranteed an acceptable minimum bandwidth allocation.

30. With respect to claims 12 and 29, Ignatius teaches the invention described in claim 11, including a method for reducing processor cycles required to send data over a communication link in packets having a packet size, the method comprising: sending a write call, from an application running in a first context, comprising a first destination and pointing to a first quantity of data stored in virtual memory destined for the first destination (Ignatius, col. 15, line 55 – col. 16, line 15) and a second destination and pointing to a second quantity of data stored in virtual memory destined for the second destination (Ignatius, col. 16, lines 16-27), both the first quantity of data and the second quantity of data greater than said packet

size to a driver (Ignatius, col. 16, lines 46-60), a network interface controller (Ignatius, Fig. 1, elements 76 and 78; col. 4, line 62 – col. 5, line 13), running in a second context through a socket (Ignatius, col. 16, lines 46-60); and generating, by the driver, a plurality of first packets less than or equal to the packet size destined for the first destination from the first quantity of data and the plurality of second packets less than or equal to the packet size destined for the second destination from the second quantity of data (Ignatius, col. 16, lines 46-60), the first and second quantities of data located at the translated physical memory locations (Ignatius, col. 3, lines 57-61).

Ignatius does not explicitly teach performing a zero-copy write.

However, Ganger teaches performing a zero-copy write translating the virtual memory locations for the first and second quantities of data to physical memory locations (Ganger, page 71, 2nd full paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ignatius in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

The combination of Ignatius and Ganger does not explicitly teach use of a multimedia data file.

However, Haddock teaches a method wherein the multimedia data file requires real-time delivery (Haddock, col. 8, lines 45-47).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the combination of Ignatius and Ganger in view of Haddock in order to enable transmitting at least one packet to the second destination before transmitting a second packet for the first destination. One would be motivated to do so in order to employ a weighted fair queuing delivery schedule which shares available bandwidth so that high priority traffic is usually sent first, but low priority traffic is still guaranteed an acceptable minimum bandwidth allocation.

31. With respect to claims 13 and 30, Ignatius teaches the invention described in claim 1, including a method for reducing processor cycles required to send data over a communication link in packets having a packet size, the method comprising: sending a write call, from an application running in a first context, comprising a first destination and pointing to a first quantity of data stored in virtual memory destined for the first destination (Ignatius, col. 15, line 55 – col. 16, line 15) and a second destination and pointing to a second quantity of data stored in virtual memory destined for the second destination (Ignatius, col. 16, lines 16-27), both the first quantity of data and the second quantity of data greater than said packet size to a driver (Ignatius, col. 16, lines 46-60), a network interface controller (Ignatius, Fig. 1, elements 76 and 78; col. 4, line 62 – col. 5, line 13), running in a second context through a socket (Ignatius, col. 16, lines 46-60); and generating, by the driver, a plurality of first packets less than or equal to the packet size destined for the first destination from the first quantity of data and the plurality of second packets less than or equal to the packet size destined for the second destination from the second quantity of data (Ignatius, col. 16, lines 46-60), the first and second quantities of data located at the translated physical memory locations (Ignatius, col. 3, lines 57-61).

Ignatius does not explicitly teach performing a zero-copy write.

However, Ganger teaches performing a zero-copy write translating the virtual memory locations for the first and second quantities of data to physical memory locations (Ganger, page 71, 2nd full paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ignatius in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

The combination of Ignatius and Ganger does not explicitly teach use of a multimedia data file.

However, Haddock teaches a method wherein the multimedia data file is a video file, an audio file, or a game file (Haddock, col. 8, lines 45-47).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the combination of Ignatius and Ganger in view of Haddock in order to enable transmitting at least one packet to the second destination before transmitting a second packet for the first destination. One would be motivated to do so in order to employ a weighted fair queuing delivery schedule which shares available bandwidth so that high priority traffic is usually sent first, but low priority traffic is still guaranteed an acceptable minimum bandwidth allocation.

32. With respect to claims 14 and 31, Ignatius teaches the invention described in claim 1, including a method for reducing processor cycles required to send data over a

communication link in packets having a packet size, the method comprising: sending a write call, from an application running in a first context, comprising a first destination and pointing to a first quantity of data stored in virtual memory destined for the first destination (Ignatius, col. 15, line 55 – col. 16, line 15) and a second destination and pointing to a second quantity of data stored in virtual memory destined for the second destination (Ignatius, col. 16, lines 16-27), both the first quantity of data and the second quantity of data greater than said packet size to a driver (Ignatius, col. 16, lines 46-60), a network interface controller (Ignatius, Fig. 1, elements 76 and 78; col. 4, line 62 – col. 5, line 13), running in a second context through a socket (Ignatius, col. 16, lines 46-60); and generating, by the driver, a plurality of first packets less than or equal to the packet size destined for the first destination from the first quantity of data and the plurality of second packets less than or equal to the packet size destined for the second destination from the second quantity of data (Ignatius, col. 16, lines 46-60), the first and second quantities of data located at the translated physical memory locations (Ignatius, col. 3, lines 57-61).

Ignatius does not explicitly teach performing a zero-copy write.

However, Ganger teaches performing a zero-copy write translating the virtual memory locations for the first and second quantities of data to physical memory locations (Ganger, page 71, 2nd full paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ignatius in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

The combination of Ignatius and Ganger does not explicitly teach use of a multimedia data file.

However, Haddock teaches a method wherein said quantity of data comprises at least a portion of a file having a format chosen from the group of formats consisting of MPEG-1, MPEG-2, MPEG-4, H.264, MP3, QuickTime, AVI, Audio/Video, real-time data in RTP format, and combinations thereof (Haddock, col. 8, lines 45-47).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the combination of Ignatius and Ganger in view of Haddock in order to enable transmitting at least one packet to the second destination before transmitting a second packet for the first destination. One would be motivated to do so in order to employ a weighted fair queuing delivery schedule which shares available bandwidth so that high priority traffic is usually sent first, but low priority traffic is still guaranteed an acceptable minimum bandwidth allocation.

33. With respect to claims 35 and 43, Ignatius teaches a method for reducing buffering requirements on a network switch; the method comprising: receiving a write call from an application running in a first context through a socket (Ignatius, col. 6, line 48 – col. 7, line 9), the write call comprising a plurality of destinations, including a first destination and a second destination (Ignatius, col. 15, line 55 – col. 16, line 15), and pointing to a first quantity of data destined for the first destination (Ignatius, col. 15, line 65 - col. 16, line 15), and pointing to a second quantity of data destined for the second destination (Ignatius, col. 16, lines 16-27); packetizing said first quantity of data into a plurality of packets less than or

equal to a packet size, each packet destined for the first destination (Ignatius, col. 16, lines 46-60); generating at least one packet comprising at least a portion of the second quantity of data destined for the second destination (Ignatius, col. 17, lines 12-29); transmitting a first packet destined for the first destination to the network switch (Ignatius, col. 12, lines 45-52).

Ignatius does not explicitly teach use of a write call received at a driver running a second context.

However, Ganger teaches the write call received at a driver running a second context (Ganger, page 54, 2.2 Application-Level Networking, 1st paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ignatius in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

The combination of Ignatius and Ganger does not explicitly teach transmitting at least one packet to the second destination before transmitting a second packet for the first destination.

However, Haddock teaches transmitting the at least one packet destined for the second destination to the network switch, before transmitting a second packet destined for the first destination (Haddock, col. 11, lines 31-36 and 62 – col. 12, line 8).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the combination of Ignatius and Ganger in view of Haddock in order to enable transmitting at least one packet to the second destination before transmitting a second packet for the first destination. One would be motivated to do so in order to employ a weighted fair queuing delivery schedule which shares available bandwidth so that high priority traffic is usually sent first, but low priority traffic is still guaranteed an acceptable minimum bandwidth allocation.

34. With respect to claim 36, Ignatius teaches the invention described in claim 35, including a method for reducing buffering requirements on a network switch; the method comprising: receiving a write call from an application running in a first context through a socket (Ignatius, col. 6, line 48 – col. 7, line 9), the write call comprising a plurality of destinations, including a first destination and a second destination (Ignatius, col. 15, line 55 – col. 16, line 15), and pointing to a first quantity of data destined for the first destination (Ignatius, col. 15, line 65 - col. 16, line 15), and pointing to a second quantity of data destined for the second destination (Ignatius, col. 16, lines 16-27); packetizing said first quantity of data into a plurality of packets less than or equal to a packet size, each packet destined for the first destination (Ignatius, col. 16, lines 46-60); generating at least one packet comprising at least a portion of the second quantity of data destined for the second destination (Ignatius, col. 17, lines 12-29); transmitting a first packet destined for the first destination to the network switch (Ignatius, col. 12, lines 45-52).

Ignatius does not explicitly teach use of a write call received at a driver running a second context.

However, Ganger teaches the write call received at a driver running a second context (Ganger, page 54, 2.2 Application-Level Networking, 1st paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ignatius in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

The combination of Ignatius and Ganger does not explicitly teach transmitting at least one packet to the second destination before transmitting a second packet for the first destination.

However, Haddock teaches transmitting the at least one packet destined for the second destination to the network switch, before transmitting a second packet destined for the first destination (Haddock, col. 11, lines 31-36 and 62 – col. 12, line 8).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the combination of Ignatius and Ganger in view of Haddock in order to enable transmitting at least one packet to the second destination before transmitting a second packet for the first destination. One would be motivated to do so in order to employ a weighted fair queuing delivery schedule which shares available bandwidth so that high priority traffic is usually sent first, but low priority traffic is still guaranteed an acceptable minimum bandwidth allocation.

The combination of Ignatius and Ganger does not explicitly teach transmitting at least one packet to the second destination before transmitting a second packet for the first destination.

However, Haddock teaches transmitting the at least one packet destined for the second destination to the network switch, before transmitting a second packet destined for the first destination (Haddock, col. 11, lines 31-36 and 62 – col. 12, line 8) and a method wherein the first quantity of data comprises at least a portion of a video media file (Haddock, col. 8, lines 45-47).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the combination of Ignatius and Ganger in view of Haddock in order to enable transmitting at least one packet to the second destination before transmitting a second packet for the first destination. One would be motivated to do so in order to employ a weighted fair queuing delivery schedule which shares available bandwidth so that high priority traffic is usually sent first, but low priority traffic is still guaranteed an acceptable minimum bandwidth allocation.

35. With respect to claim 37, Ignatius teaches the invention described in claim 35, including a method further comprising generating a plurality of said packets comprising at least a portion of the second quantity of data destined for the second destination (Ignatius, col. 17, lines 12-29).

36. With respect to claim 38, Ignatius teaches the invention described in claim 35, including a method further comprising further comprising: transmitting the second packet destined for the first destination (Ignatius, col. 16, lines 16-27).

37. With respect to claim 40, Ignatius teaches the invention described in claim 35, including a method further comprising: communicating at least one of said packets to a network interface card (Ignatius, Fig. 1, elements 76 and 78; col. 4, line 62 – col. 5, line 13).

38. With respect to claim 41, Ignatius teaches the invention described in claim 35, including a method for reducing buffering requirements on a network switch; the method comprising: receiving a write call from an application running in a first context through a socket (Ignatius, col. 6, line 48 – col. 7, line 9), the write call comprising a plurality of destinations, including a first destination and a second destination (Ignatius, col. 15, line 55 – col. 16, line 15), and pointing to a first quantity of data destined for the first destination (Ignatius, col. 15, line 65 - col. 16, line 15), and pointing to a second quantity of data destined for the second destination (Ignatius, col. 16, lines 16-27); packetizing said first quantity of data into a plurality of packets less than or equal to a packet size, each packet destined for the first destination (Ignatius, col. 16, lines 46-60); generating at least one packet comprising at least a portion of the second quantity of data destined for the second destination (Ignatius, col. 17, lines 12-29); transmitting a first packet destined for the first destination to the network switch (Ignatius, col. 12, lines 45-52).

Ignatius does not explicitly teach use of a write call received at a driver running a second context.

However, Ganger teaches the write call received at a driver running a second context (Ganger, page 54, 2.2 Application-Level Networking, 1st paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Ignatius in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

The combination of Ignatius and Ganger does not explicitly teach transmitting at least one packet to the second destination before transmitting a second packet for the first destination.

However, Haddock teaches transmitting the at least one packet destined for the second destination to the network switch, before transmitting a second packet destined for the first destination (Haddock, col. 11, lines 31-36 and 62 – col. 12, line 8).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the combination of Ignatius and Ganger in view of Haddock in order to enable transmitting at least one packet to the second destination before transmitting a second packet for the first destination. One would be motivated to do so in order to employ a weighted fair queuing delivery schedule which shares available bandwidth so that high priority traffic is usually sent first, but low priority traffic is still guaranteed an acceptable minimum bandwidth allocation.

The combination of Ignatius and Haddock does not explicitly teach performing a zero-copy write.

However, Ganger teaches a method wherein the write call points to the first and second quantity of data stored in virtual memory and further comprising: performing a zero-copy write translating the virtual memory locations for the first and second quantities of data to physical memory locations (Ganger, page 71, 2nd full paragraph).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the combination of Ignatius and Haddock in view of Ganger in order to enable performing a zero-copy write. One would be motivated to do so in order to reduce redundancy (most notably, repeated data copying), both in work and in memory usage.

39. With respect to claim 42, Ignatius teaches the invention described in claim 35, including a method wherein said write call comprises a write vector (Ignatius, col. 15, lines 11-13).

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Alicia Baturay whose telephone number is (571) 272-3981. The examiner can normally be reached at 7:30am - 5pm, Monday - Thursday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Pwu can be reached on (571) 272-6798. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Alicia Baturay
March 15, 2010

/Jeffrey Pwu/
Supervisory Patent Examiner, Art Unit 2446